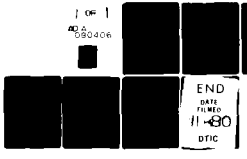
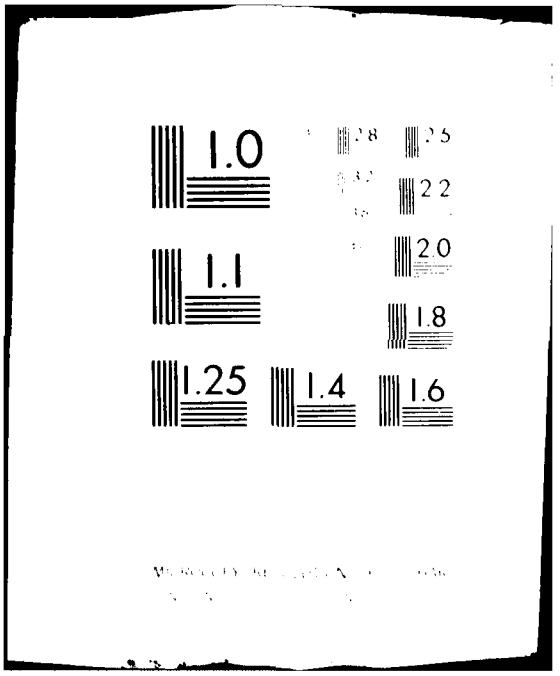
AD-A090 406 ARMY COMMUNICATIONS RESEARCH AND DEVELOPMENT COMMAND --ETC F/G 9/2
CODING AND PROCESSING FOR RELIABLE DATA TRANSMISSION,(U)
JUN 80 W A HUBER

UNCLASSIFIED NL

1 OF 1
AD A090406




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2

CODING AND PROCESSING FOR RELIABLE DATA TRANSMISSION(U)

JUN 1980

MR. WILLIAM A. HUBER
US ARMY COMMUNICATIONS RESEARCH
AND DEVELOPMENT COMMAND
FORT MONMOUTH, NEW JERSEY 07703

The coding and processing techniques discussed in this paper are designed to improve the reliability of digital message reception in the presence of noise. High noise conditions occur quite frequently during radio reception especially when these signals originate at forward area military positions. The usual cause for this poor radio reception is related to the constraints imposed by low transmitter power, limits on air time, and unfavorable terrain conditions. The importance of improving the reliability of forward area message reception is stressed because the information so obtained provides essential descriptors that are pertinent to command and control functions.

In an effort to determine the cause of message delivery failure in the presence of increasing noise, laboratory tests were performed under controlled noise conditions. The results of these tests indicated that as the signal-to-noise power ratio decreased errors first appeared in the data character framing information. It should be noted that the message synchronizing procedure used to decode received digital data is hierarchical in nature involving bit synchronization, data character framing, and message block identification. If any one of these functions is not correctly implemented the transmited message cannot be successfully decoded. Ideally, as increased noise degrades the data communications channel the synchronizing system should not fail before the message information becomes unintelligible; a condition that is presently not satisfied because data character framing requires error-free code reception.

The reason for the premature failure of the data character framing function in the presence of increasing transmission channel

noise lies in the inherent weakness of the method used to establish data character framing. Data character framing is accomplished by template matching techniques where a given bit sequence is transmitted during the message preamble that must be perfectly matched at the receiver terminal. With this method a single bit error will prevent the correct identification of data character framing thereby causing loss of the complete message. The vulnerability of data character framing to noise cannot be reduced by employing error correction coding because subsequent error correction decoding requires a priori knowledge of data character framing.

The identification of data character framing as the most noise vulnerable function in the data communications synchronizing procedure led to the development of a data character framing technique that does not use noise sensitive template matching. This technique involves the introduction of a new code and associated data processing system. Before describing the new system, techniques used to obtain bit synchronization will be discussed.

II BIT SYNCHRONIZATION

The following brief discussion of bit synchronization is provided to illustrate that this function can be maintained under noise conditions that are far more severe than can be tolerated by the present data character extraction process. By increasing the performance of the data character framing process in the presence of noise, the weak link in the data reception reliability chain can be strengthened.

Figure 1 illustrates a typical data message structure. Bit synchronization data consists of a series of 1's and 0's. This cyclic signal structure is convenient for obtaining the timing information necessary to synchronize the receiver clock with the message signal. Receiver synchronization is usually accomplished by means of a Phase-Locked Loop (PLL) such as illustrated in Figure 2. In its basic form, the PLL (1) consists of a phase detector, a loop filter/amplifier and a receiver clock which consists of a voltage controlled oscillator (VCO). The phase detector is used to compare the phase of the incoming signal with that produced by the VCO, and then generate a resultant error voltage proportional to the difference. The loop filter provides short term memory and usually contains a linear amplifier. Amplification is provided so that oscillator control voltage can be derived from minimal error voltages thus maintaining close tracking between the incoming signal and the VCO. The VCO is designed to have a nominal frequency stability within a few cycles of that of the incoming signal frequency. In operation this frequency differential

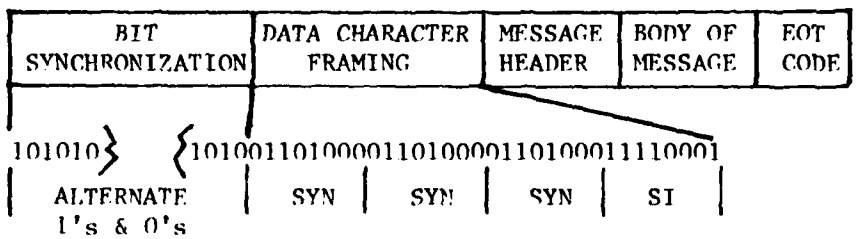


Figure 1. Typical data message structure.

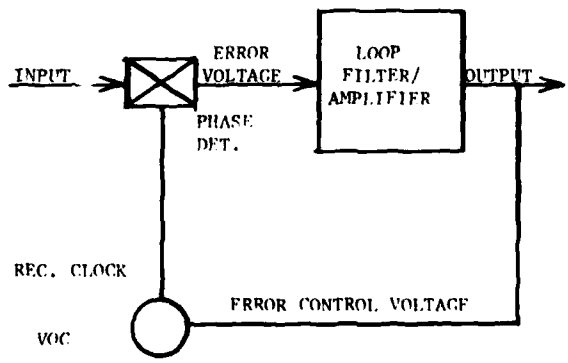


Figure 2. Phase-locked loop.

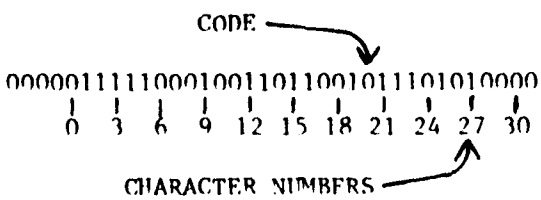


Figure 3. Proposed data character framing code.

(309)

is used to derive an error control voltage that drives the frequency of the VCO into lock with the incoming signal.

The PLL's superior performance during high noise reception conditions is related to the following: The PLL acts as an extremely narrowband filter which rejects sideband noise. It does this by supplying a local coherent signal that enhances its ability to extract narrowband signals from noise. The memory provided by the PLL supplies additional flywheel stability to the VCO during the presence of random noise.

III DATA CHARACTER FRAMING CODE

A typical 32-bit data character framing code illustrated in Figure 1, consists of three American Standard Code for Information Interchange (ASCII) "SYN" characters, and one ASCII "SI" character. It should be noted that the weakness in the present method for extracting data character framing information is not related to the ASCII code characters utilized, but is related to the inability of the technique used to cope with bit errors. This weakness is illustrated by the following: In operation the receiver contains a replica of the 32-bit data character framing code which it attempts to match with the continuous flow of a 32-bit data sequence encoded in the incoming signal stream. When a 32-bit match is obtained it is used to establish the timing period for the start of the first bit of the first data character of the message. The start of successive data characters is then identified by means of a local counter driven by the bit synzhronized clock. It is readily seen from the above procedure that a single bit error in the received data character framing code will cause the loss of the transmitted message.

It is emphasized that the present technique for processing the data character framing code is a serial bit-by-bit procedure with no tangible results until the entire code has been processed. Not until this point is reached can a GO or NO-GO decision be made with respect to the starting period for message data character framing. It would therefore be advantageous to have a technique that would provide the required data character framing by sampling less than the complete 32-bit code. This feature would reduce the chance of error in direct proportion to the reduction of the bit sample size. It would also be advantageous to make the samples sufficiently small so that a multiplicity of samples could be processed during one 32-bit data character framing period. Such a technique would be especially effective during noisy reception conditions as it would provide several opportunities to obtain the required error-free bit sample necessary to establish

message data character framing.

A data character framing code that possesses the noise immunity features discussed above has been developed, based on viewing data framing as a countdown rather than a pattern. This code is illustrated in serial form in Figure 3. The pertinent noise immunity features that make this code ideally suited to the data character framing application are: it contains a high information density, it provides for simultaneous parallel and serial scanning, it identifies error-free bit sequences, and it establishes data character framing from only one small error-free bit sequence within the frame.

The high information density of the new code can be illustrated by first examining its structure. This is done with reference to Figures 4 and 5. The code is derived from the 32, 5-bit character code shown tabulated in paralled form in Figure 4. For identification purposes numerals 0 - 31 have been arbitrarily assigned to the codes as indicated in Figure 4. It will be noted from Figure 4 that successive characters of the code are formed by shifting each of the bits in columns 1,2,3 and 4, one bit position to the left and judiciously selecting bits for the vacated column one positions. If the 5-bit code shown in Figure 4 were to be used for randomly transmitting numerals 0 through 31, its efficiency with respect to information density would be no better than if the standard 5-bit binary code had been used. However, the data character framing operation is not a random function; instead, it is a precisely defined countdown procedure. Present data character framing techniques cannot take advantage of this countdown feature because no method is available for identifying the progression of the countdown process.

The structure of the data character framing code shown in Figure 3 contains the necessary data to meet this countdown criteria. That is, as the countdown progresses each new bit must contain the additional data necessary to identify the position of that bit within the countdown sequence. To demonstrate this, it should be noted that all the information contents of the 32, 5-bit characters of Figure 4 reside in the 36 bits comprising row 0 and column 1. All the remaining 124 bits in columns 2,3,4 and 5 are redundant. This can be seen by reference to Figure 4, where for any 5 by 5-bit array, the bits contained in the bottom row and left column of the array are identical. This structural characteristic was used to derive the data character framing code illustrated in Figure 3, which consists of the four 0 bits of row 1, plus the 32 bits of column 1. If this sequence of bits is scanned with a 5-bit parallel aperture, then at any point in the scanning process the bits within the aperture identify the

(311)

Character Number	Bit Number	Character Number	Bit Number
	54321		54321
0	00000	16	11011
1	00001	17	10110
2	00011	18	01100
3	00111	19	11001
4	01111	20	10010
5	11111	21	00101
6	11110	22	01011
7	11100	23	10111
8	11000	24	01110
9	10001	25	11101
10	00010	26	11010
11	00100	27	10101
12	01001	28	01010
13	10011	29	10100
14	00110	30	01000
15	01101	31	10000

Figure 4. Proposed code data structure.

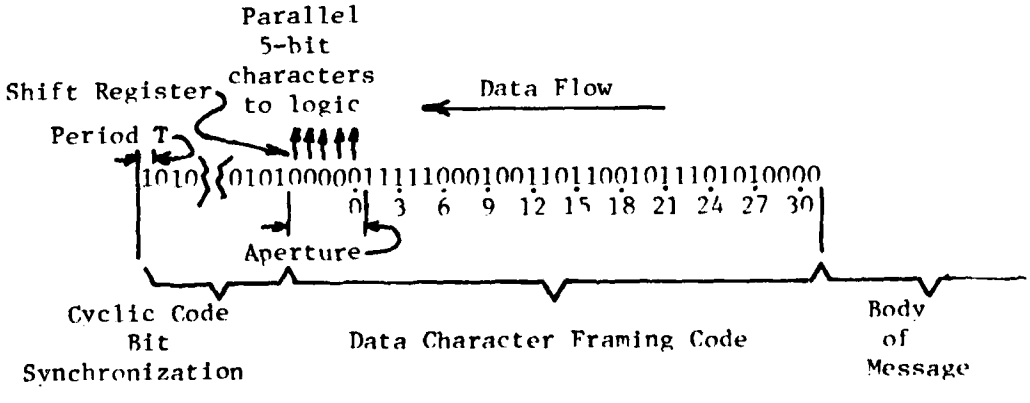


Figure 5. Processing the data character framing code.

312

position of the aperture within the code. To establish data character framing code countdown it is only necessary to pass the input message data stream through a 5-bit shift register and extract the 5-bit countdown codes from the parallel output of this shift register. A simplified schematic of this procedure is illustrated in Figure 5.

In the above discussion of data character framing code countdown it was tacitly assumed that the received code contained no errors. If the received code contains errors then the associated extracted countdown code will also be in error. An example of the countdown procedure where the signal stream contains errors is illustrated in Figure 6. When referring to Figure 6, it should be noted that for simplicity of illustration it is assumed that the data signal stream is stationary and that the 5-bit aperture scans from left to right. The bit identification numbering system used in Figures 3 and 5 is retained in Figure 6. This numbering system is also consistant with that of Figure 4 when it is considered in conjunction with the corresponding character in the scanning 5-bit aperture.

No generality will be lost if we start the discussion on code processing by assuming that the 5-bit scanning aperture is reading code (00000) corresponding to bit number "0" on the 0-31 scale. For the present we will assume that the first six bits of the code are error-free. As the 5-bit aperture scans one bit position to the right code (00001) corresponding to bit number "1" will be read. Up to this point the two codes read have been in consecutive order, namely: "0" and "1". If it is now assumed that bit number "2" is in error, as is indicated in Figure 6 by underlining the bit in error, then this code will be read as character number "10". Because of the symetrical nature of the code count, this abrupt jump in count order can only be attributed to an error in either or both codes forming the discontinuous transition. At this point it is not necessary to locate this error; it is sufficient to know that an error exists thus making the data unreliable for processing.

As the 5-bit aperture continues to scan the right, it will be noted from Figure 6 that the effect of the error is prevalent for the five bit numbers "10","21","22","23", and "24". From this example it can be seen that the minimum number of 5-bit aperture samples required to establish an error-free countdown sample set is six. That is, if six successive 5-bit aperture scans result in the generation of six consecutive bit numbers, then this sample set is error-free and the last bit of this set can be considered correct and used as a reference to specify the number of bit periods remaining before the start of the first character of the body of the message. That this is true can be seen by noting that the set of six, 5-bit aperture samples

313

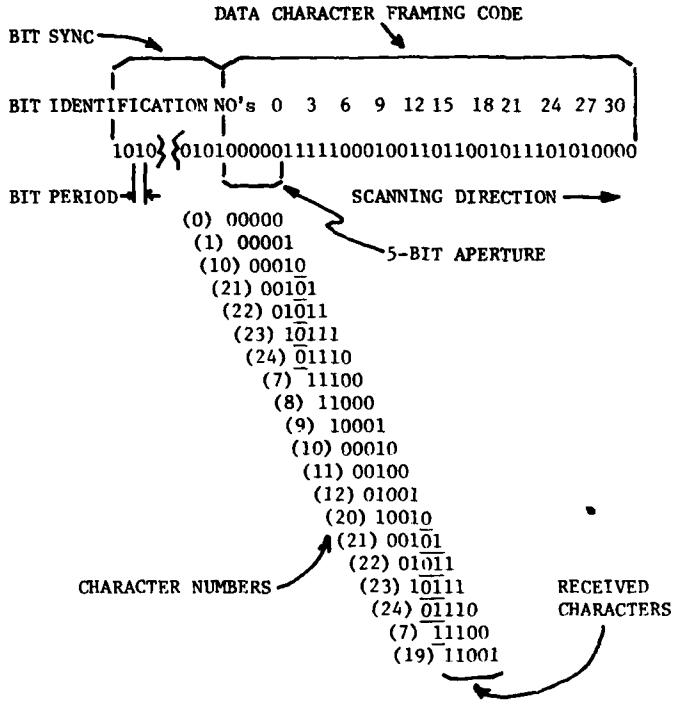


Figure 6. Countdown procedure during error conditions.

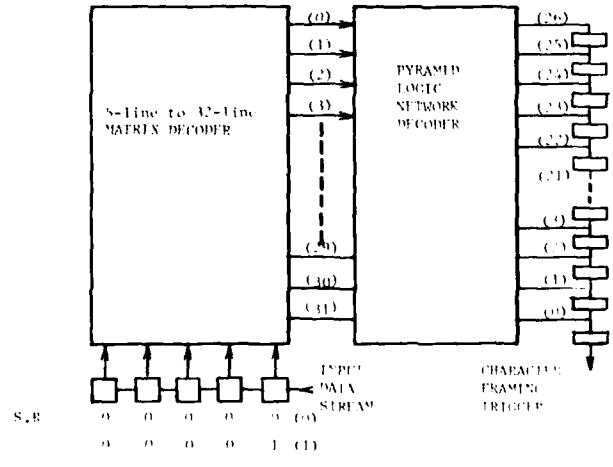


Figure 8. Functional diagram for processing the data character framing code.

(314)

indicated by bit numbers "7","8","9","10","11" and "12" contain no discontinuity of count. Bit number "12" of this sample set can therefore be used to initiate a countdown of the remaining 19-bit periods of the data character framing code. The completion of the 19-bit countdown marks the beginning of the first character of the body of the message. This information can then be used to synchronize an internal counter for use in maintaining data character framing, message block identification, and error detection and correction.

IV DATA CHARACTER FRAMING LOGIC

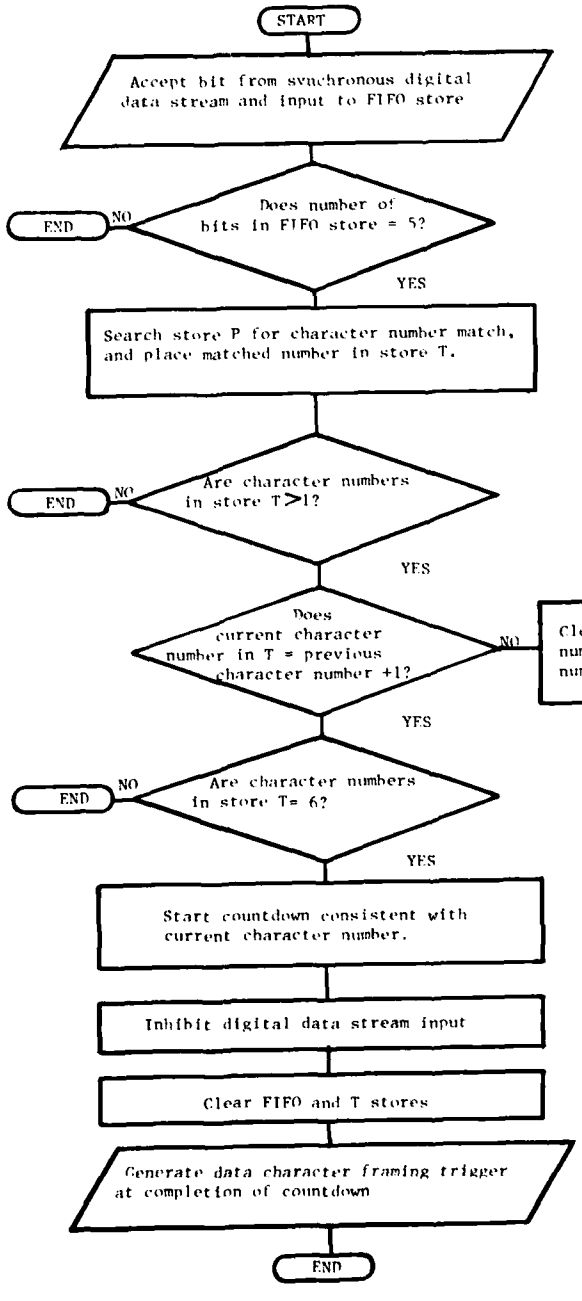
A flowchart illustrating the requirements for processing the data character framing code is contained in Figure 7. The input processing requirement is for a 5-bit store of the first-in-first-out (FIFO) type. This store must operate at the synchronous transmission rate, and be capable of accepting a binary digital data stream, while providing a synchronous 5-bit parallel output.

A functional diagram of a system for firmware implementation of the data character framing code is presented in Figure 8. The FIFO store of Figure 7 is implemented in Figure 8 by a 5-bit shift register. The data signal input is shifted through the register at a synchronous rate by the internal clock. The 5-bit parallel output of the shift register is fed to a 5-line to 32-line matrix decoder. A more detailed look at a portion of the 5-line to 32-line matrix decoder is shown in Figure 9. Input lines of the matrix decoder are activated by a logical-1; output lines of the matrix decoder respond by producing a logical-1.

A simplified partial schematic of the pyramid logic network decoder is shown in Figure 10. The function of this logic is to filter out all random inputs; producing a single output only in response to six consecutive inputs. The input latches (2) of Figure 10 provide the necessary data storage so that during each bit period a constant logical-1 or logical-0 is applied to the associate AND gates. In Figure 10 it is assumed that consecutive character number codes (0),(1),(2),(3),(4) and (5) have been received. This places a logical-1 on the pyramiding AND gates resulting in a logical-1 appearing at the output of AND gate "E_1", where it is applied to the input of a 26 stage countdown shift register. The output of the shift register initiates data character framing.

An additional feature not shown in Figure 10 that must be provided for is the ability to automatically reset to logical-0 any latch that has been set to logical-1, but does not belong to a consecutive set of six character numbers. Reset logic to accomplish

(315)



Maximum processing time must be less than the bit interval.

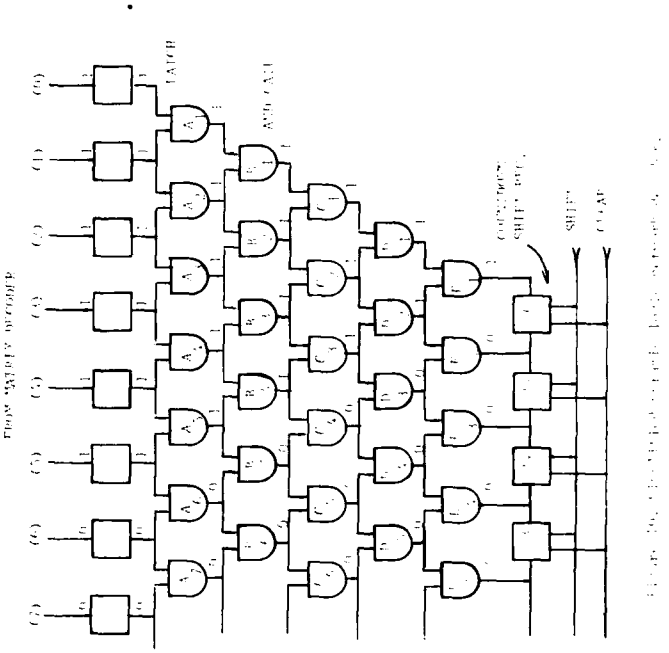
P is permanent store that contains the 32,5-bit code characters.

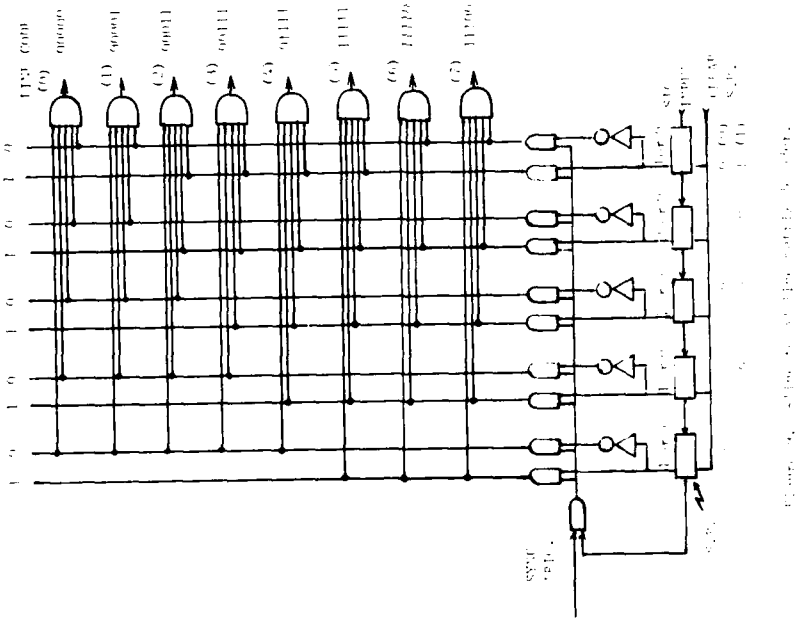
T is temporary store.

NOTE: After END's system remains quiescent until activated by next synchronous trigger.

Figure 7. Flowchart for processing the data character framing code.

(316)





(317)

this is shown in Figure 11, and the associate time logic of operational events is tabulated in Table I. Only three input lines from the matrix decoder are illustrated in Figure 11 as the operation of the remaining 29 lines is identical to those presented. Referring to Table I, the horizontal lines represent circuit activating triggers with time delays increasing when going from the top to bottom of the columns. The column numbers identify circuit nodes in Figure 11.

V CONCLUSIONS

A new binary digital code and associated data preocessing technique has been described that is capable of extracting data character framing information from a noise contaminated digital data stream. The need for such an approach has been emphasized because of the inherent weakness of the present method for extracting data character framing, which is based on a noise sensitive template matching technique. The ineffectiveness of the present method for obtaining data character framing was substantiated during simulated operational tests where, as the reception conditions deterioriated because of increased noise, the data character framing function was first to fail. Further tests indicated that if the ability to extract data character framing could be extended down in the noise to a point where the message contents become unintelligible, message delivery rates could be significantly increased. The structured code of the proposed data character framing technique satisfies the above condition by virtue of its ability to operate in noisy environments. It accomplishes this desirable result without resorting to increased transmitter powers or receiver sensitivities.

The effectiveness of the proposed data character framing method is indicated by noting that the system's noise immunity features are such that it is only necessary to receive ten consecutive error-free bits (six consecutive character numbers) to establish correct data character framing. In the limit, all remaining bits in the data character framing code could be in error. These ten consecutive error-free bits may occur anywhere within the data character framing message preamble period.

The magnitude of improved reliability in message delivery that can be realized by using the proposed data character framing technique is related to the reduction in the size of the data sample. If a S/N power ratio of 7.5 dB is assumed, then the probability of receiving an error as read from the curve of Figure 12, (3) is 10^{-2}, or on the average of one error per hundred bits. Present data character framing requires a string of 32 error-free bits, so there is a

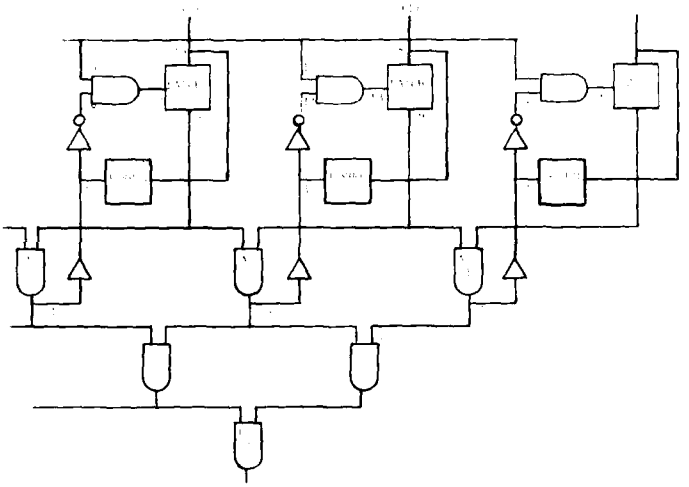


TABLE I

Time logic table

		(1) DATA		(1) 1-SHOT	(1) RESET				(2) DATA		(2) 1-SHOT	(2) RESET				(3) DATA		(3) 1-SHOT	(3) RESET			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
RESET	t_1	0	0	0/0	1	1	1	0	0	0	0/0	1	1	1	0	0	0	0/0	1	1	1	0
DATA	t_2	1	1	1/0	0	0	0	0	0	0	0/0	1	0	0	0	0	0	0/0	1	0	0	0
1-SHOT	t_3	0	1	1/0	0	0	0	0	0	0	0/0	1	0	0	0	0	0	0/0	1	0	0	0
RESET	t_4	0	1	1/0	0	1	0	0	0	0	0/0	1	1	1	0	0	0	0/0	1	1	1	0
DATA	t_5	0	1	1/1	0	0	0	1	1	1	1/0	0	0	0	0	0	0	0/0	1	0	0	0
1-SHOT	t_6	0	1	0/1	0	0	0	1	0	1	1/0	0	0	0	0	0	0	0/0	1	0	0	0
RESET	t_7	0	1	0/1	0	1	0	1	0	1	1/0	0	1	0	0	0	0	0/0	1	1	1	0
DATA	t_8	0	1	0/1	0	0	0	1	0	1	1/1	0	0	0	1	1	1	1/0	0	0	0	0
1-SHOT	t_9	0	1	0/1	0	0	0	1	0	1	0/1	0	0	0	1	0	1	1/0	0	0	0	0
RESET	t_{10}	0	1	0/1	0	1	0	1	0	1	0/1	0	1	0	1	0	1	1/0	0	1	0	0
DATA	t_{11}	0	1	0/1	0	0	0	1	0	1	0/1	0	0	0	1	0	1	1/1	0	0	0	1
1-SHOT	t_{12}	0	1	0/1	0	0	0	1	0	1	0/1	0	0	0	1	0	1	0/1	0	0	0	1
RESET	t_{13}	0	1	0/1	0	1	0	1	0	1	0/1	0	1	0	1	0	1	0/1	0	1	0	1

probability of success of 0.67. The proposed data character framing technique requires a string of only 10 error-free bits, so there is a probability of success of 0.89, which results in 33% improvement in message delivery.

To adapt the proposed data character framing technique to present military data communications systems all that is required is to substitute, in the message preamble, the new data character framing code for the presently used data character framing code. The new code is then processed by methods such as have been described. These methods could be implemented by either software or firmware techniques. Firmware implementation could be in the form of an integrated circuit chip. Both the effectiveness of the data character framing method and its simplicity of implementation suggest its universal use as a standard method for obtaining data character framing.

ACKNOWLEDGEMENTS

The author wishes to acknowledge his indebtedness to Messrs. David K. Ruppe and Thomas J. Wheeler, USA CORADCOM, for reviewing the manuscript and providing suggestions for improving its contents. The author would also like to thank Mr. Richard E. Loiselle, USA CORADCOM for his efforts during the early development of the data character framing code concept.

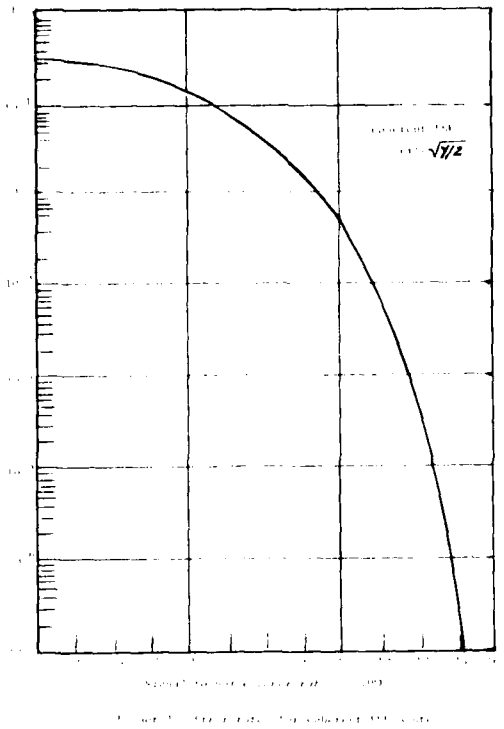
REFERENCES

1. Gardner, Floyd M., "Phaselock Techniques", 2nd Ed., John Wiley & Sons, Inc., N. Y. 1979.

2. Peatman, John B., "The Design of Digital Systems", McGraw-Hill Book Co., N. Y., 1972.

3. Stein, S. and Jones, J. J., "Modern Communications Principles", McGraw-Hill Book Co., N. Y., 1967.

(320)



APPENDIX

$P_e = 1/2 \operatorname{erfc}\sqrt{\gamma/2}$ Probability of error

The complementary error function is defined by

$$\operatorname{erfc} x = 1 - \operatorname{erf}$$

$$= \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt$$

where

$$\gamma = \frac{u^2}{2N}$$ *Power ratio*

N = average noise power level at filter output at instant of sampling

u = signal level at filter output momentarily containing the data bit at the same instant of sampling N.

The above analysis (3) applies to a Frequency Shift Keying (FSK) signal using synchronous (coherent) detection. This type of detection requires an exact replica of each of the frequencies representing the binary states to be available at the receiver. To satisfy this requirement PLL's are used. As the signal and noise add in the filters, it is expected that the filter output with the signal will be algebraically larger than the filter without the signal. If this is not true and error is indicated.